



(19)

Europäisches Patentamt

European Patent Office

Office européen des brevets



(11)

EP 0 803 789 A2

(12)

EUROPEAN PATENT APPLICATION

(43) Date of publication:
29.10.1997 Bulletin 1997/44

(51) Int Cl. 6: G06F 1/00

(21) Application number: 97302490.4

(22) Date of filing: 11.04.1997

(84) Designated Contracting States:
DE ES FR GB SE(72) Inventor: Benson, Glenn Stuart
81739 Munich (DE)

(30) Priority: 26.04.1996 GB 9608696

(74) Representative: Guyatt, Derek Charles
Intellectual Property Department
International Computers Limited
Cavendish Road
Stevenage, Herts, SG1 2DY (GB)(71) Applicant: EUROPEAN COMPUTER-INDUSTRY
RESEARCH CENTRE GmbH
D-81925 München (DE)

(54) Software copy protection mechanism

(57) A copy protection mechanism for protecting software against copying, consists of a challenge mechanism embedded in each protected item of software. The challenge mechanism has no access to the customer's private keying material. In operation, the challenge mechanism sends a random challenge to the customer's signature server. The signature server signs the challenge, using the customer's private keying material

and then returns the signed challenge to the challenge mechanism. The challenge mechanism then verifies the signed challenge, using the customer's public keying material, and prohibits the customer from using some or all of the protected item of software unless the verification is successful. The mechanism permits every customer to receive an identical copy of the copy protected program with the embedded challenge mechanism.

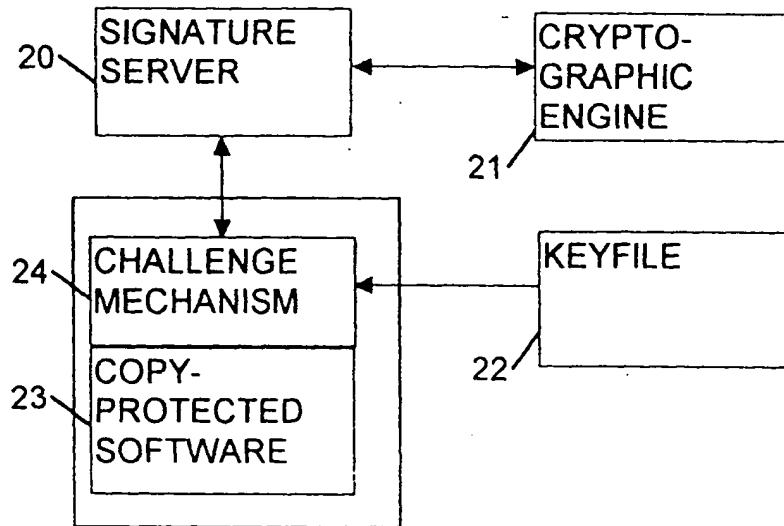


FIG. 2

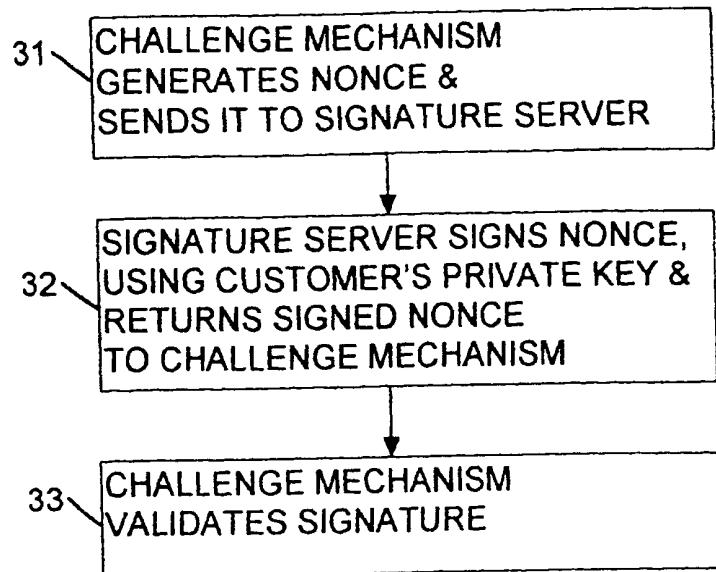


FIG. 3

DescriptionBackground to the Invention

This invention relates to electronic copy protection mechanisms for protecting software against unauthorised copying.

The Business Software Alliance estimates the 1995 financial losses attributed to software piracy as US\$8.1 Billion for business application software and US\$15.2 Billion for all software. Solutions have been proposed in two areas: (i) improved Intellectual Property Rights (IPR) legislation, and (ii) enhanced electronic copy protection (ECP) mechanisms. IPR legislation and enforcement are improving in many countries, but there are still significant difficulties in other parts of the world. As a result, some vendors are currently reassessing ECP.

It is desirable for any ECP mechanism to satisfy the following requirements.

- The ECP mechanism should prohibit unauthorized users from executing copy protected software.
- The ECP mechanism should not prohibit the user from making backups.
- The ECP mechanism should make only standard hardware and software assumptions. For example, although hardware dongles provide excellent copy protection services, many vendors do not wish to limit the sale of the software to the collection of users who own or are willing to install a dongle.
- The ECP mechanism should have minimal impact upon the user interface. The visible impact should be limited to the customer's initial login to the operating system and/or smart card. Subsequent impact upon the user interface should be relegated to relatively minor performance concerns.
- The ECP mechanism should not limit execution of the copy protected software to a limited collection of machines. When a customer legitimately purchases software, the customer should be able to execute the software on any machine regardless of ownership. The customer should optionally be able to authorize simultaneous execution of the software on multiple machines.
- The ECP mechanism should have no required network dependencies in order to execute an already purchased copy protected program.
- The vendor should be permitted to distribute an identical version of the copy protected software to all users. This requirement permits the copy protected software to be distributed through normal channels such as, for example, CD-ROMs, floppy

disks, or network bulletin boards.

- It should be excessively difficult and/or computationally infeasible for a potential software pirate to circumvent the copy protection mechanism without modifying the copy protected program. This requirement serves as an important virus-protection measure because a digital signature supplied by the vendor would not validate if a pirate distributes a modified version of the original program.
- The ECP mechanism should not compromise any of the customer's private keying material. In particular, the ECP mechanism should not disclose the customer's private keying material to the vendor, any program produced by the vendor, or any simple Trojan horse program. While the primary functionality of copy protection is to protect the software vendor, one must not do so at the expense of the customer.
- The ECP mechanism should be available in either a software-only version or a hardware-assisted (smart card) version, to assure widespread market acceptance.
- The least time consuming attack by a potential software pirate should be byte-code disassembly of the copy protected software. In order to thwart the copy protection mechanism, the pirate must remove or change the ECP.

Choudhury et al. ["Copyright Protection for Electronic Publishing over Computer Networks", available as at 27th March 1996 on Word Wide Web at <http://ftp.research.att.com/dist/anoncc/copyright.epub.ps.Z>] propose a mechanism in which a protected document can be viewed only via a specially configured viewer program, which allows a user to view the document only if the user supplies the viewer with the user's private keying material. This deters the user from distributing unauthorized copies of the viewer program, since that would require the user to divulge his or her private keying material to others. However, because Choudhury's mechanism requires that the viewer program obtain access to the customer's private keying material, it breaks one of the requirements listed above. Furthermore, Choudhury's mechanism may not be used in conjunction with a smart card that is configured to avoid releasing private keying material.

The object of the present invention is to provide an improved ECP mechanism that is able to satisfy the above requirements.

The ECP mechanism of the present invention makes use of asymmetric cryptography, also known as public key cryptography. In asymmetric cryptography, each user has public keying material and private keying material. Each user may post his or her public keying

material to a publicly accessed directory without compromising the corresponding private keying material. Normally, the user guards the private keying material as a close secret. Using the RSA asymmetric encryption algorithm, for example, a pair of users may encrypt and then subsequently decrypt a message using either of two methods: (i) encrypt using the public keying material and decrypt using the private keying material; or (ii) encrypt using the private keying material and decrypt using the public keying material. Two examples are presented below.

Secret message: A user, Alice, posts her public keying material to a well-known directory or bulletin board. A second user, Bob, wishes to send a confidential message to Alice. Bob encrypts the message using Alice's public keying material and sends the encrypted message to Alice. Since Alice is the only user with access to the corresponding private keying material, only Alice may decrypt the message to discover its original content.

Digital signature: A digital signature is an electronic analog of a handwritten signature. After posting her public keying material, Alice encrypts a message using the private keying material. Since anyone may access the public keying material, there is no message secrecy. However, since Alice is the only user with access to the private keying material, no one else can "forge Alice's signature" by performing the encryption. Any user may validate Alice's signature using the public keying material.

Both examples depend upon the fact that Alice closely guards her private keying material. Otherwise, the cryptographic system may neither guarantee secrecy nor ensure signature validity. The best known mechanism for protecting one's private keying material is through the use of a smart card. In this case, the smart card is a device with no interface for releasing private keying material (in a non-cryptographically protected form).

Although smart cards provide the best protection, social factors of electronic commerce may provide a role in ensuring private keying material protection. One of the significant difficulties associated with asymmetric encryption services is authentication. For example, if Alice posts her public keying material to a public directory, then how does Bob assess validity? That is, a pirate may attempt to masquerade as Alice but post the pirate's keying material. Some commercial organizations are beginning to provide solutions to this problem by acting as Certification Authorities (CA). For (possibly) a fee, the CA solicits identifying material from potential customers such as a driver's license or passport. After validating the identifying material, the CA posts the customer's public keying material to a public directory, and the CA signs a certificate that holds the customer's public keying material. Standardized services, for example, X.500 may be adopted to help facilitate the use of directories that contain public keying material.

Once a user posts his or her public keying material to the CA, the user will probably make an extensive effort to protect his or her private keying material. In this case, if the user's private keying material were to become unknowingly compromised, then the user would have cause for significant concern because networked vendors may authorize electronic commerce transactions based on the information found in the public directory.

10

Summary of the Invention

According to the invention there is provided a computer system comprising a copy protection mechanism for protecting software against copying, the copy protection mechanism comprising challenge means associated with a protected item of software, and response means in which a customer's private keying material is securely stored, wherein:

20

(a) the challenge means has no access to the customer's private keying material;

25

(b) the response means comprises means for signing information using the customer's private keying material and then returning the signed information to the challenge means, and

30

(c) the challenge means comprises means for verifying said signed information, using the customer's public keying material, and for prohibiting the customer from using some or all of said item of software unless said verification is successful.

Brief Description of the Drawings

Figure 1 is a flow diagram of a purchasing protocol used when a customer wishes to purchase software that is protected by an ECP mechanism in accordance with the invention.

Figure 2 is a block diagram showing the software components that are required to be installed in the customer's machine to enable the customer to run the copy protected software.

Figure 3 is a flow diagram showing the operation of the ECP mechanism in the protected software.

Figure 4 shows the format of a nonce (random number expressed in hexadecimal form) generated by the ECP mechanism.

Figure 5 is a flowchart showing the operation of a random number generator used to generate nonces.

Description of an Embodiment of the Invention

One ECP mechanism in accordance with the invention will now be described by way of example with reference to the accompanying drawings.

Purchasing protocol

Figure 1 shows a purchasing protocol used when a customer wishes to purchase software that is protected by an ECP mechanism in accordance with the invention. It is assumed that each party (i.e. the vendor and each potential customer) has its own public and private keying material. Each party makes its public keying material available to other parties, but keeps its private keying material secret.

In step 1, the customer obtains the protected software from a vendor. The customer may get the software, for example, on a floppy disk or CD-ROM at a PC store. Alternatively, the customer may download the software from a network bulletin board. A challenge mechanism, to be described later, is embedded in protected software in such a way that a potential attacker cannot easily separate the challenge mechanism from the protected program. The attacker would need to disassemble the code and to manually remove the challenge mechanism. The challenge mechanism has the vendor's public keying material embedded in it. As will be described, the challenge mechanism prevents the user from running the software at this stage.

In step 2, the customer sends a registration package to the software vendor, for example by electronic mail. This registration package contains a reference to a public directory that holds the customer's public keying material.

In step 3, the software vendor embeds the customer's public keying material into a keyfile and sends the keyfile to the customer, for example by electronic mail. Once the customer installs the keyfile, the ECP mechanism permits the customer to execute the software.

The creation of the keyfile by the vendor is performed by a keyfile generator, which is a program that executes at the vendor's facility. The vendor must take care to guard this program because it accesses the vendor's private keying material. In use of the keyfile generator, an operator enters the following information:

Vendor name:	The name of the vendor's company.
Vendor password:	The password that unlocks the vendor company's private keying material (decrypt the private keying material using the password). Company employees who do not know the password cannot generate keyfiles.
Customer name:	The name of a customer for whom to generate a keyfile. The name indexes into a database of public keying material.
Keyfile name:	The name of a new keyfile.

After obtaining this information, the keyfile generator builds a keyfile, containing the customer's public key. The keyfile appears to the customer as a completely random sequence of values. Building of the keyfile involves the following operations. First, the keyfile generator creates a file and inserts the customer's public keying material into the file, along with thousands of decoy bits. In the present example, each keyfile contains approximately 480,000 decoy bits. This number of bits represents a significant amount of decoy material, and can fit into a standard e-mail message. Each keyfile stores the customer's public keying material in a different location. Additionally, each keyfile has encrypted customer information embedded in it without disclosing the required encryption key. This encrypted customer information permits a software vendor to easily identify the owner of a keyfile in the event that the keyfile appears in a public location such as a bulletin board. The keyfile generator then encrypts and re-encrypts the keyfile multiple times, using different algorithms. Finally, the keyfile generator signs the keyfile using the vendor's private keying material.

Customer software

Figure 2 shows the software components that are required to be installed in the customer's machine (computer) to enable the customer to run the copy protected software. These consist of a signature server 20, and a cryptographic engine 21. Also shown are the keyfile 22 and the copy protected software 23. As mentioned above, the copy protected software includes a challenge mechanism 24.

The signature server is a program that the customer trusts, which the customer executes when the system initially boots. The customer enables the system by inserting a floppy disk that contains an encrypted copy of the customer's private keying material. The signature server then prompts the customer for a pass phrase used to decrypt the floppy. The signature server then executes in the background waiting for signature requests.

It should be noted that the signature server never releases the customer's private keying material out of its process boundary. The signature server relies on operating system protections to ensure its own integrity. The signature server executes in its own address space and communicates with external processes. The signature server stores the customer's password in wired memory (images of the memory never appear in swap space).

The cryptographic engine is a dynamic linked library (DLL) available in both the customer's and the vendor's home country, that performs cryptographic services. The signature server links with the cryptographic engine. Services provided by the cryptographic engine include asymmetric and symmetric encryption and a one-way hash function.

Operation of ECP mechanism

Figure 3 shows the operation of the ECP mechanism. This is performed when the user initially attempts to execute the protected software, and is also repeated periodically during execution.

(Box 31) The challenge mechanism of the protected software generates an unguessable nonce (random number) and passes the nonce to the signature server with a signature request.

(Box 32) When it receives the nonce, the signature server first checks that the nonce presented to it corresponds exactly to the format Figure 4, with the exception of the 32 random-appearing characters. If it does not, the signature server denies the signature request. Assuming that the nonce is in the correct format, the signature server uses the cryptographic engine to sign the nonce using the customer's private keying material. The signature server then returns the signed nonce to the challenge mechanism in the protected software.

(Box 33) When it receives the signed nonce, the challenge mechanism accesses the keyfile associated with the protected software and calls a signature validation function in the challenge mechanism to validate the vendor's signature of the keyfile, using the vendor's public keying material that is embedded in the challenge mechanism. This validation of the keyfile signature ensures that an attacker cannot modify the keyfile or its digital signature without additionally modifying the challenge mechanism. Vendors may optionally augment this protection using additional proprietary lines of defence. If the keyfile has been modified, the challenge mechanism hangs the program.

Assuming the signature is validated, the challenge mechanism then parses the keyfile, using a proprietary, vendor-specific algorithm, to locate the customer's public keying material in the keyfile, and extracts this public keying material. The challenge mechanism then calls its signature validation function, to validate the digital signature computed over the nonce using the customer's public keying material. If the signature is not valid, the challenge mechanism hangs the program. Thus, it can be seen that the software continues executing if and only if the customer possesses the proper keyfile.

Nonce generator

Generation of a nonce is performed by a nonce generator included in the challenge mechanism. Operation of the nonce generator is as follows. First, the nonce generator queries a large number of system parameters, e.g., the system time, the amount of space remaining free in the page table, the number of logical disk drives, the names of the files in the operating system's directory, etc.

Next, the nonce generator builds a random number, using a random number generator. The random number generator consists of two process threads, referred to

herein as Thread 1 and Thread 2. Figure 5 shows the operation of Thread 1, which is the main thread of the random number generator.

(Box 51) Thread 1 first creates a data structure value_list, for holding a list of counter values. The list is initially empty.

(Box 52) Thread 1 sets a current counter value to zero, and sets a done_test flag to FALSE.

(Box 53) Thread 1 then forks Thread 2. Thread 2 posts an asynchronous disk access, and then sleeps until the disk access is complete. When the disk access is complete, Thread 2 sets the done_test flag to TRUE. Note that Thread 1 and Thread 2 share the done_test flag.

(Box 54) Thread 1 increments the counter value by one.

(Box 55) Thread 1 then tests whether the done_test flag is now TRUE, indicating that the disk access initiated by Thread 2 is complete. If done_test is FALSE, the process returns to box 54. Thus it can be seen that, while waiting for the disk access to complete, Thread 1 continually increments the counter value.

(Box 56) When done_test is TRUE, Thread 1 terminates Thread 2, and saves the counter value in the first free location in value_list.

(Box 57) Thread 1 then calls a Statstest function, which estimates the degree of randomness of the counter values saved on value_list. This function may use, for example, the Chi-Square, Kolmogorov-Smirnov, and Serial Correlation Tests. (The Serial Correlation Test is described in D.Knuth, The Art of Computer Programming, Addison-Wesley Publishing Co, Reading MA, 2nd Edition, 1981). The Statstest function may be optimized to ensure that complicated calculations are not repeated for each disk access. Statstest returns a value which indicates how many low-order bits of each saved counter value should be considered random.

(Box 58) Thread 1 compares the value returned by Statstest when combined with the length of the value_list with a predetermined threshold value, to determine whether enough random bits have now been generated. If not enough random bits have been generated, the process returns to box 52 above, so as to generate and save another counter value.

(Box 59) When the required number of random bits has been generated, Thread 1 extracts the specified number of low-order bits from each counter value in the list, and returns this sequence of bits as the output random number.

In summary, it can be seen that the random number generator exploits the unpredictability in the timing of a series of disk accesses as a source of randomness in the generation of nonces. (See P.Fenstermacher et al., "Cryptographic randomness from air turbulence in disk drives", Advances in Cryptology: Crypto '94, pages 114-120, Springer-Verlag, 1994). By forking new threads on each disk access, the random number generator also exploits unpredictabilities in the operation of

the operating system's scheduler as a second source of randomness.

The analysis performed by the Statstest function permits the random number generator to self-tune for any speed processor and disk, by computing the number of low-order bits of each saved counter value to return. For example, a system with a high-variance disk access time will generate more random bits per-disk access than a system with a low-variance disk access time. For example, for a Quantum 1080s disk (6ms average write time), and a 486 66MHz processor, the system generates approximately 45 bits per second. Alternatively, one may hard code the number of bits per-disk access and use a de-skewing technique to ensure a good degree of randomness.

The nonce generator also queries the operating system to ensure that it posts each disk access to an actual disk. The final output nonce is formed by combining the output random number from the random number generator with the result of querying the system parameters as described above.

The nonce generator described above works best when executing on an operating system that provides direct access to the disk, e.g., Windows 95 or Windows NT. In such an operating system, special operating system calls available to programs executing in user space permit a program to bypass the operating system's internal buffering mechanism and write directly to the disk. Most programs do not take advantage of these special operating system calls because they may be relatively inefficient and difficult to use. On Windows 95 and Windows NT, for example, a program may only use these special calls if the program accesses data that is a multiple of the disk's sector size. The program may discover the sector size by querying the operating system.

If the operating system does not provide direct access to the disk, then the challenge mechanism could still use the disk timing random number generator. However, in this case, the quality of the generated values would have a greater reliance upon unpredictabilities in the operating system's scheduler as opposed to the variance inherent to the disk access time.

The example of the invention described above assumes that the operating system permits a program to fork multiple threads within a single address space. Additionally, the example of the invention assumes that the operating system permits the threads to access synchronization variables such as semaphores. Most modern operating systems (including Windows 95 and Windows NT) provide these services. The example of the invention uses multiple threads to implement a mechanism which quantifies each disk access time. However, if an implementation of the invention were to execute on a system that does not provide multiple threads or synchronization variables, then the nonce generator could substitute other mechanisms, e.g. querying a physical clock.

Attacks and defences

The ECP mechanism described above is robust because it protects itself against attack as described below.

1) Disclosed private keying material

In this form of attack, a pirate legitimately purchases the software and then discloses both the software and the pirate's private keying material to an unauthorised user.

The ECP mechanism deters this form of attack because the electronic commerce market provides many incentives for guarding one's private keying material. In order to ensure that a pirate does not generate keying material for the exclusive use of copy protection, the software vendor should consult a CA to ensure that each customer has pre-authorized his or her private keying material to be used for multiple purposes.

The ECP mechanism does not prohibit a customer from maintaining multiple asymmetric key pairs. However, at least one key pair could be used in multiple ways. Perhaps, for example, a software vendor may choose to use the same public keying material both (i) to validate a customer's signature of the electronic transaction used to purchase the software and (ii) to build the customer's keyfile.

2) Certificate revocation

In this form of attack, immediately after receiving a keyfile from the vendor, the pirate issues a certificate revocation package (CRP). The purpose of the CRP is to announce publicly that the pirate's private keying material has been compromised. Subsequently, the CAs add the pirate's CRP to their certificate revocation lists. The infrastructure encourages electronic commerce participants to disregard any certificate mentioned in a certificate revocation list. At this point the pirate freely distributes his or her private keying material in order to thwart the copy protection mechanism.

The ECP mechanism deters this form of attack because a pirate who wishes to generate a significant profit from breaking ECP has little incentive to mount a CRP attack. The problem is that the pirate probably wishes to remain anonymous in order to avoid legal prosecution. Although the pirate may plead innocence by claiming that someone stole his or her private keying material, the pirate should assume that he or she may be a prime suspect in a potential criminal investigation launched by a law enforcement agency.

A law enforcement investigation might not be a primary concern to a pirate who wishes to generate only a minor profit. However, we hope that the time, trouble, and expense, associated with issuing a CRP outweighs the benefit of circumventing the copy protection for the benefit of only a few close friends.

3) Clandestine keyfile

In this form of attack, the pirate generates a new public/private keying material pair. Next, the pirate creates a clandestine keyfile that contains the new public keying material. Alternatively, the pirate modifies an original keyfile by substituting the new public keying material. The pirate distributes the copy protected software, the clandestine keyfile, and the new keying material.

The ECP mechanism protects against this type of attack using three lines of defence. The first line of defence is the digital signature of the keyfile, described above. The digital signature ensures that, if a pirate modifies the keyfile, the challenge mechanism does not validate the signature. The pirate may potentially thwart this line of defence by changing the implementation of the challenge mechanism, by substituting a new embedded public key. Such a change to the challenge mechanism breaks an important virus-protection measure that may be used by a legitimate vendor -- signing the program in order to ensure that the program is not modified in the distribution process. Note that a vendor does not release its private keying material. This line of defence provides significant keyfile protection because it requires a pirate to disassemble the copy protected program in order to locate the vendor's public keying material.

The second line of defence is the decoy bits in the keyfile. Although the keyfile stores no confidential information, the vendor-specific algorithm for parsing a keyfile is a secret. However, a pirate may potentially learn the secret by disassembling the challenge mechanism's executable image.

The third line of defence is the multiple encryption of the keyfile. If one assumes that the pirate cannot break the encryption algorithms themselves, the pirate must disassemble the copy protected program in order to determine the required cryptographic algorithms.

Of course, no two vendors should use the three lines of defence in exactly the same way. So, if a pirate were able to generate clandestine keyfiles for one version of a program, then the pirate's keyfile-generator would not generate clandestine keyfiles for other vendor's program versions.

4) Repeated nonces

In this form of attack, the pirate forces the challenge mechanism to generate the same nonce upon each invocation. All of the pirate's customers also force their copy of the challenge mechanism to generate the same nonce. The pirate breaks the copy protection mechanism by freely distributing his or her signature of this special nonce. This attack does not require that the pirate disclose his or her private keying material.

The ECP mechanism implements multiple lines of defence against the repeated nonce attack. The first line of defence is the way in which the challenge mechanism

generates an input to the nonce generator by querying a large number of system parameters, as described. This defence is simple but is probably sufficient to thwart the casual pirate. A casual pirate probably would not be

5 willing to trouble him or herself with the difficult procedures required to break this line of defence. For example, the pirate may need to install libraries that spoof the real operating system and fool the challenge mechanism into obtaining the wrong system parameters. In
10 other words, this line of defence assumes that each customer has slightly different system parameters and as a result ensures that no two customers generate the same nonces.

The second line of defence is the sophisticated random number generator as described. However, a sufficiently crafty pirate may potentially install a library that fools both the challenge mechanism and the underlying operating system by substituting a RAM-disk for a real disk. The challenge mechanism protects against this by
15 forking new threads, which has the effect of taking into account unpredictabilities in the operating system's scheduler, as described above.

The third line of defence is the embedded encrypted customer information in the keyfile. The purpose of the
25 encrypted customer information is to permit a software vendor to easily identify the owner of a keyfile in the event that the keyfile appears in a public location such as a bulletin board. Law enforcement agencies may potentially use the customer information in a criminal investigation.

5) Shared keying material

In this form of attack, the pirate executes a copy protected program using legitimate keying material. After
35 the copy protected program boots, the pirate reuses the keying material for other purposes on other machines. The pirate never terminates the copy protected program.

40 The ECP mechanism protects against this form of attack by periodically posting a nonce and then hanging, waiting for a reply, as described.

6) Trojan Horse propagation

45 In this form of attack, a pirate vendor distributes a copy protected program that contains an embedded Trojan Horse. The Trojan Horse steals the unsuspecting customer's private keying material and transmits the private keying material back to the pirate vendor. The pirate vendor uses the stolen private keying without the legitimate user's knowledge.

50 The ECP mechanism defends against this form of attack because, as described above, the signature server never releases the private keying material out of its process boundary. The customer enables the system by inserting a floppy disk that contains an encrypted copy of the customer's private keying material. However, no

Trojan Horse software may use the information stored on the floppy disk because the Trojan Horse cannot decrypt the floppy.

7) Networked signature server

In this form of attack, a pirate organization establishes a server that signs all nonces. The pirate organization distributes the copy protected software and the keyfile. Additionally, the pirate organization establishes a network connection to its signature server. Each of the pirate's customers executes the program by sending each signature request to the pirate's signature server. Since the keyfile's public key matches the signature produced by the pirate's signature server, the pirate's customers can execute the program.

A pirate who wishes to recoup a significant profit would probably not mount this form of attack. In this case, the pirate would be required to compromise anonymity by widely distributing his or her network address.

A pirate who wishes to recoup only a minor profit has little incentive to mount this form of attack because the cost of operating and maintaining a signature server outweighs the potential benefit of defeating the copy protection mechanism for the benefit of a few close friends. A company, however, may potentially have enough financial incentive to implement a signature server on its private network. In this case, the company defeats the copy protection mechanism by condoning a violation of copyright laws.

8) Signature server vulnerability

A pirate may choose to launch an attack on the signature server rather than the copy protection mechanism. Although the signature server signs all randomly generated nonces, the customer does not wish the signature server to grant all such requests. For example, a customer would not wish the signature server to sign a request of the form "Please withdraw funds from my account".

The defence to this attack is that the format of any nonce presented to the signature server must correspond exactly to Figure 4 with the exception of the 32 random-appearing characters. If a pirate were to remove, for example, any of the three warnings, then the signature server would deny the signature request. If the pirate were to remove any of the warnings of the signed nonce, then the signature would not validate.

Some possible modifications

It will be appreciated that many modifications may be made to the system described above without departing from the scope of the present invention.

In particular, a smart card may be used to store the customer's public and private keying material. In such a smart card-enabled configuration, a pirate cannot ex-

tract the private keying material from the smart card, which provides even greater defence against attack.

In addition to implementing copy protection, the mechanism described above may be used to trace software pirates. The mechanism provides excellent traceability, without requiring that each customer obtain a unique version of the program.

The signature server, challenge mechanism, and copy protected software described above may be deployed in a number of different configurations. For example:

- 15 (i) The signature server, the challenge mechanism, and the copy protected software may all reside in the same address space on a single machine.
- (ii) The signature server may reside in one address space and the challenge mechanism and copy protected software may reside in a different address space on a single machine.
- 20 (iii) The signature server may reside in an address space on one machine and the challenge mechanism and copy protected software may reside in a different address space on a different machine.

25 Furthermore, multiple users, each with their own copy of the protected item of software, may share a common signature server, which responds to challenges from all these copies. Another alternative is that multiple users may share common private keying material. A company may use one or more of these deployment options, for example, when constructing multi-user licensing functionality.

30 In the embodiment described above, the challenge mechanism uses a random number generator to ensure freshness of the protocol. The freshness property ensures that an attacker cannot thwart the system's security guarantees by replaying old messages. Alternatively, different protocols may be used to ensure freshness. For example, in one such alternative protocol, the signature server generates a current timestamp and signs the timestamp. The challenge mechanism validates both freshness of the timestamp and correctness of the signature, and halts execution of the program if a threshold time limit is exceeded without receiving a valid signed timestamp.

35 In another possible modification, the keyfile may contain hidden information concerning selective activation of services of the copy protected program. For example, the keyfile may specify that the copy protected program may permit execution of a Print service but disable execution of a Save-On-Disk service. As another example, the keyfile may contain an expiration date that describes the last date that a particular service may execute. The copy protected program would read the keyfile to determine the services that the program should execute. A customer could obtain the ability to execute more services by requesting another keyfile from the vendor.

Claims

1. A computer system comprising a copy protection mechanism for protecting software against copying, the copy protection mechanism comprising challenge means associated with a protected item of software, and response means in which a customer's private keying material is securely stored, wherein:

(a) the challenge means has no access to the customer's private keying material;

(b) the response means comprises means for signing information using the customer's private keying material and then returning the signed information to the challenge means, and

(c) the challenge means comprises means for verifying said signed information, using the customer's public keying material, and for prohibiting the customer from using some or all of said item of software unless said verification is successful.

2. A computer system according to Claim 1 wherein said challenge means comprises means for issuing a random challenge, and wherein said information comprises said random challenge.

3. A computer system according to Claim 1 or 2 wherein said challenge means is embedded in said protected item of software.

4. A computer system according to any preceding claim wherein the system includes a keyfile for holding the customer's public keying material.

5. A computer system according to any preceding claim, including a plurality of protected items of software, each having its own challenge means, and a single response means, shared between all of said protected items.

6. A method of distributing software to a plurality of customers wherein each customer has a computer system according to any preceding claim, and wherein every customer receives an identical copy of said copy protected program and of said challenge means.

7. A computer system comprising:

(a) means for inputting a program to be copy-protected, and for embedding a challenge means in that program; and

(b) means for inputting a plurality of units of

5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
180
185
190
195
200
205
210
215
220
225
230
235
240
245
250
255
260
265
270
275
280
285
290
295
300
305
310
315
320
325
330
335
340
345
350
355
360
365
370
375
380
385
390
395
400
405
410
415
420
425
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500
505
510
515
520
525
530
535
540
545
550
555
560
565
570
575
580
585
590
595
600
605
610
615
620
625
630
635
640
645
650
655
660
665
670
675
680
685
690
695
700
705
710
715
720
725
730
735
740
745
750
755
760
765
770
775
780
785
790
795
800
805
810
815
820
825
830
835
840
845
850
855
860
865
870
875
880
885
890
895
895
900
905
910
915
920
925
930
935
940
945
950
955
960
965
970
975
980
985
990
995
1000
1005
1010
1015
1020
1025
1030
1035
1040
1045
1050
1055
1060
1065
1070
1075
1080
1085
1090
1095
1100
1105
1110
1115
1120
1125
1130
1135
1140
1145
1150
1155
1160
1165
1170
1175
1180
1185
1190
1195
1200
1205
1210
1215
1220
1225
1230
1235
1240
1245
1250
1255
1260
1265
1270
1275
1280
1285
1290
1295
1300
1305
1310
1315
1320
1325
1330
1335
1340
1345
1350
1355
1360
1365
1370
1375
1380
1385
1390
1395
1400
1405
1410
1415
1420
1425
1430
1435
1440
1445
1450
1455
1460
1465
1470
1475
1480
1485
1490
1495
1500
1505
1510
1515
1520
1525
1530
1535
1540
1545
1550
1555
1560
1565
1570
1575
1580
1585
1590
1595
1600
1605
1610
1615
1620
1625
1630
1635
1640
1645
1650
1655
1660
1665
1670
1675
1680
1685
1690
1695
1700
1705
1710
1715
1720
1725
1730
1735
1740
1745
1750
1755
1760
1765
1770
1775
1780
1785
1790
1795
1800
1805
1810
1815
1820
1825
1830
1835
1840
1845
1850
1855
1860
1865
1870
1875
1880
1885
1890
1895
1900
1905
1910
1915
1920
1925
1930
1935
1940
1945
1950
1955
1960
1965
1970
1975
1980
1985
1990
1995
1995
2000
2005
2010
2015
2020
2025
2030
2035
2040
2045
2050
2055
2060
2065
2070
2075
2080
2085
2090
2095
2100
2105
2110
2115
2120
2125
2130
2135
2140
2145
2150
2155
2160
2165
2170
2175
2180
2185
2190
2195
2200
2205
2210
2215
2220
2225
2230
2235
2240
2245
2250
2255
2260
2265
2270
2275
2280
2285
2290
2295
2300
2305
2310
2315
2320
2325
2330
2335
2340
2345
2350
2355
2360
2365
2370
2375
2380
2385
2390
2395
2400
2405
2410
2415
2420
2425
2430
2435
2440
2445
2450
2455
2460
2465
2470
2475
2480
2485
2490
2495
2500
2505
2510
2515
2520
2525
2530
2535
2540
2545
2550
2555
2560
2565
2570
2575
2580
2585
2590
2595
2600
2605
2610
2615
2620
2625
2630
2635
2640
2645
2650
2655
2660
2665
2670
2675
2680
2685
2690
2695
2700
2705
2710
2715
2720
2725
2730
2735
2740
2745
2750
2755
2760
2765
2770
2775
2780
2785
2790
2795
2800
2805
2810
2815
2820
2825
2830
2835
2840
2845
2850
2855
2860
2865
2870
2875
2880
2885
2890
2895
2900
2905
2910
2915
2920
2925
2930
2935
2940
2945
2950
2955
2960
2965
2970
2975
2980
2985
2990
2995
2995
3000
3005
3010
3015
3020
3025
3030
3035
3040
3045
3050
3055
3060
3065
3070
3075
3080
3085
3090
3095
3100
3105
3110
3115
3120
3125
3130
3135
3140
3145
3150
3155
3160
3165
3170
3175
3180
3185
3190
3195
3200
3205
3210
3215
3220
3225
3230
3235
3240
3245
3250
3255
3260
3265
3270
3275
3280
3285
3290
3295
3300
3305
3310
3315
3320
3325
3330
3335
3340
3345
3350
3355
3360
3365
3370
3375
3380
3385
3390
3395
3400
3405
3410
3415
3420
3425
3430
3435
3440
3445
3450
3455
3460
3465
3470
3475
3480
3485
3490
3495
3500
3505
3510
3515
3520
3525
3530
3535
3540
3545
3550
3555
3560
3565
3570
3575
3580
3585
3590
3595
3600
3605
3610
3615
3620
3625
3630
3635
3640
3645
3650
3655
3660
3665
3670
3675
3680
3685
3690
3695
3700
3705
3710
3715
3720
3725
3730
3735
3740
3745
3750
3755
3760
3765
3770
3775
3780
3785
3790
3795
3800
3805
3810
3815
3820
3825
3830
3835
3840
3845
3850
3855
3860
3865
3870
3875
3880
3885
3890
3895
3895
3900
3905
3910
3915
3920
3925
3930
3935
3940
3945
3950
3955
3960
3965
3970
3975
3980
3985
3990
3995
3995
4000
4005
4010
4015
4020
4025
4030
4035
4040
4045
4050
4055
4060
4065
4070
4075
4080
4085
4090
4095
4100
4105
4110
4115
4120
4125
4130
4135
4140
4145
4150
4155
4160
4165
4170
4175
4180
4185
4190
4195
4200
4205
4210
4215
4220
4225
4230
4235
4240
4245
4250
4255
4260
4265
4270
4275
4280
4285
4290
4295
4300
4305
4310
4315
4320
4325
4330
4335
4340
4345
4350
4355
4360
4365
4370
4375
4380
4385
4390
4395
4400
4405
4410
4415
4420
4425
4430
4435
4440
4445
4450
4455
4460
4465
4470
4475
4480
4485
4490
4495
4500
4505
4510
4515
4520
4525
4530
4535
4540
4545
4550
4555
4560
4565
4570
4575
4580
4585
4590
4595
4600
4605
4610
4615
4620
4625
4630
4635
4640
4645
4650
4655
4660
4665
4670
4675
4680
4685
4690
4695
4700
4705
4710
4715
4720
4725
4730
4735
4740
4745
4750
4755
4760
4765
4770
4775
4780
4785
4790
4795
4800
4805
4810
4815
4820
4825
4830
4835
4840
4845
4850
4855
4860
4865
4870
4875
4880
4885
4890
4895
4895
4900
4905
4910
4915
4920
4925
4930
4935
4940
4945
4950
4955
4960
4965
4970
4975
4980
4985
4990
4995
4995
5000
5005
5010
5015
5020
5025
5030
5035
5040
5045
5050
5055
5060
5065
5070
5075
5080
5085
5090
5095
5100
5105
5110
5115
5120
5125
5130
5135
5140
5145
5150
5155
5160
5165
5170
5175
5180
5185
5190
5195
5200
5205
5210
5215
5220
5225
5230
5235
5240
5245
5250
5255
5260
5265
5270
5275
5280
5285
5290
5295
5300
5305
5310
5315
5320
5325
5330
5335
5340
5345
5350
5355
5360
5365
5370
5375
5380
5385
5390
5395
5400
5405
5410
5415
5420
5425
5430
5435
5440
5445
5450
5455
5460
5465
5470
5475
5480
5485
5490
5495
5500
5505
5510
5515
5520
5525
5530
5535
5540
5545
5550
5555
5560
5565
5570
5575
5580
5585
5590
5595
5600
5605
5610
5615
5620
5625
5630
5635
5640
5645
5650
5655
5660
5665
5670
5675
5680
5685
5690
5695
5700
5705
5710
5715
5720
5725
5730
5735
5740
5745
5750
5755
5760
5765
5770
5775
5780
5785
5790
5795
5800
5805
5810
5815
5820
5825
5830
5835
5840
5845
5850
5855
5860
5865
5870
5875
5880
5885
5890
5895
5895
5900
5905
5910
5915
5920
5925
5930
5935
5940
5945
5950
5955
5960
5965
5970
5975
5980
5985
5990
5995
5995
6000
6005
6010
6015
6020
6025
6030
6035
6040
6045
6050
6055
6060
6065
6070
6075
6080
6085
6090
6095
6100
6105
6110
6115
6120
6125
6130
6135
6140
6145
6150
6155
6160
6165
6170
6175
6180
6185
6190
6195
6200
6205
6210
6215
6220
6225
6230
6235
6240
6245
6250
6255
6260
6265
6270
6275
6280
6285
6290
6295
6300
6305
6310
6315
6320
6325
6330
6335
6340
6345
6350
6355
6360
6365
6370
6375
6380
6385
6390
6395
6400
6405
6410
6415
6420
6425
6430
6435
6440
6445
6450
6455
6460
6465
6470
6475
6480
6485
6490
6495
6500
6505
6510
6515
6520
6525
6530
6535
6540
6545
6550
6555
6560
6565
6570
6575
6580
6585
6590
6595
6595
6600
6605
6610
6615
6620
6625
6630
6635
6640
6645
6650
6655
6660
6665
6670
6675
6680
6685
6690
6695
6700
6705
6710
6715
6720
6725
6730
6735
6740
6745
6750
6755
6760
6765
6770
6775
6780
6785
6790
6795
6795
6800
6805
6810
6815
6820
6825
6830
6835
6840
6845
6850
6855
6860
6865
6870
6875
6880
6885
6890
6895
6895
6900
6905
6910
6915
6920
6925
6930
6935
6940
6945
6950
6955
6960
6965
6970
6975
6980
6985
6990
6995
6995
7000
7005
7010
7015
7020
7025
7030
7035
7040
7045
7050
7055
7060
7065
7070
7075
7080
7085
7090
7095
7095
7100
7105
7110
7115
7120
7125
7130
7135
7140
7145
7150
7155
7160
7165
7170
7175
7180
7185
7190
7195
7195
7200
7205
7210
7215
7220
7225
7230
7235
7240
7245
7250
7255
7260
7265
7270
7275
7280
7285
7290
7295
7295
7300
7305
7310
7315
7320
7325
7330
7335
7340
7345
7350
7355
7360
7365
7370
7375
7380
7385
7390
7395
7395
7400
7405
7410
7415
7420
7425
7430
7435
7440
7445
7450
7455
7460
7465
7470
7475
7480
7485
7490
7495
7495
7500
7505
7510
7515
7520
7525
7530
7535
7540
7545
7550
7555
7560
7565
7570
7575
7580
7585
7590
7595
7595
7600
7605
7610
7615
7620
7625
7630
7635
7640
7645
7650
7655
7660
7665
7670
7675
7680
7685
7690
7695
7695
7700
7705
7710
7715
7720
7725
7730
7735
7740
7745
7750
7755
7760
7765
7770
7775
7780
7785
7790
7795
7795
7800
7805
7810
7815
7820
7825
7830
7835
7840
7845
7850
7855
7860
7865
7870
7875
7880
7885
7890
7895
7895
7900
7905
7910
7915
7920
7925
7930
7935
7940
7945
7950
7955
7960
7965
7970
7975
7980
7985
7990
7995
7995
8000
8005
8010
8015
8020
8025
8030
8035
8040
8045
8050
8055
8060
8065
8070
8075
8080
8085
8090
8095
8095
8100
8105
8110
8115
8120
8125
8130
8135
8140
8145
8150
8155
8160
8165
8170
8175
8180
8185
8190
8195
8195
8200
8205
8210
8215
8220
8225
8230
8235
8240
8245
8250
8255
8260
8265
8270
8275
8280
8285
8290
8295
8295
8300
8305
8310
8315
8320
8325
8330
8335
8340
8345
8350
8355
8360
8365
8370
8375
8380
8385
8390
8395
8395
8400
8405
8410
8415
8420
8425
8430
8435
8440
8445
8450
8455
8460
8465
8470
8475
8480
8485
8490
8495
8495
8500
8505
8510
8515
8520
8525
8530
8535
8540
8545
8550
8555
8560
8565
8570
8575
8580
8585
8590
8595
8595
8600
8605
8610
8615
8620
8625
8630
8635
8640
8645
8650
8655
8660
8665
8670
8675
8680
8685
8690
8695
8695
8700
8705
8710
8715
8720
8725
8730
8735
8740
8745
8750
8755
8760
8765
8770
8775
8780
8785
8790
8795
8795
8800
8805
8810
8815
8820
8825
8830
8835
8840
8845
8850
8855
8860
8865
8870
8875
8880
8885
8890
8895
8895
8900
8905
8910
8915
8920
8925
8930
8935
8940
8945
8950
8955
8960
8965
8970
8975
8980
8985
8990
8995
8995
9000
9005
9010
9015
9020
9025
9030
9035
9040
9045
9050
9055
9060
9065
9070
9075
9080
9085
9090
9095
9095
9100
9105
9110
9115
9120
9125
9130
9135
9140
9145
9150
9155
9160
9165
9170
9175
9180
9185
9190
9195
9195
9200
9205
9210
9215
9220
9225
9230
9235
9240
9245
9250
9255
9260
9265
9270
9275
9280
9285
9290
9295
9295
9300
9305
9310
9315
9320
9325
9330
9335
9340
9345
9350
9355
9360
9365
9370
9375
9380
9385
9390
9395
9395
9400
9405
9410
9415
9420
9425
9430
9435
9440
9445
9450
9455
9460
9465
9470
9475
9480
9485
9490
9495
9495
9500
9505
9510
9515
9520
9525
9530
9535
9540
9545
9550
9555
9560
9565
9570
9575
9580
9585
9590
9595
9595
9600
9605
9610
9615
9620
9625
9630
9635
9640
9645
9650
9655
9660
9665
9670
9675
9680
9685
9690
9695
9695
9700
9705
9710
9715
9720
9725
9730
9735
9740
9745
9750
9755
9760
9765
9770
9775
9780
9785
9790
9795
9795
9800
9805
9810
9815
9820
9825
9830
9835
9840
9845
9850
9855
9860
9865
9870
9875
9880
9885
9890
9895
9895
9900
9905
9910
9915
9920
9925
9930
9935

12 to 14 wherein the keyfile includes information concerning selective activation of services of the copy protected program.

16. A computer system including a random number generator for generating random numbers by timing responses to disk accesses, wherein said random number generator is embedded in a program and runs in that program's address space. 5

10

17. A computer system according to Claim 16 wherein random number generator includes means for forking new threads in such a manner as to introduce an additional degree of randomness into said random numbers by exploiting unpredictabilities in the operating system's scheduler. 15

18. A computer system according to Claim 16 or 17, including means for performing a statistical test to determine the number of random bits obtained by each of said disk accesses, and means for causing said disk accesses to be repeated until a predetermined number of random bits has been obtained. 20

25

30

35

40

45

50

55

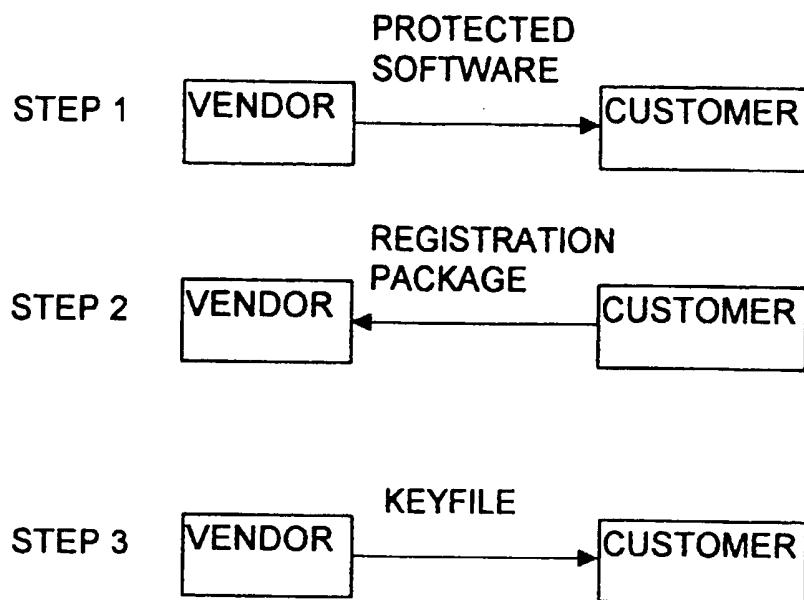


FIG. 1

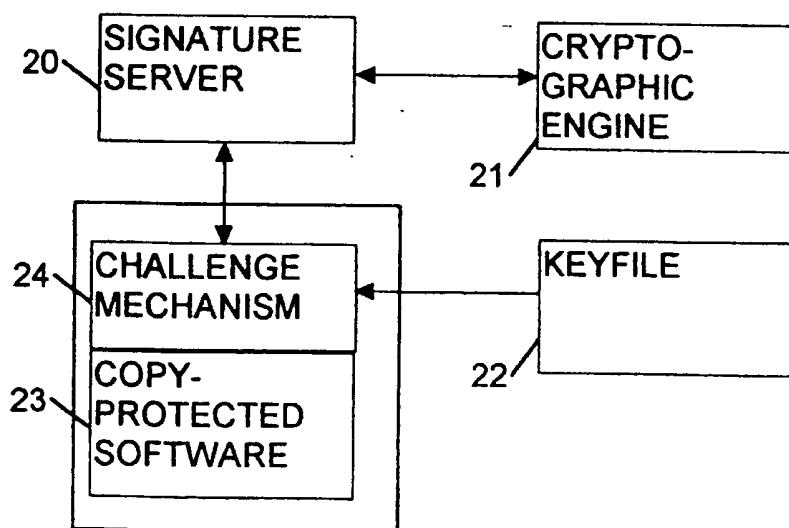


FIG. 2

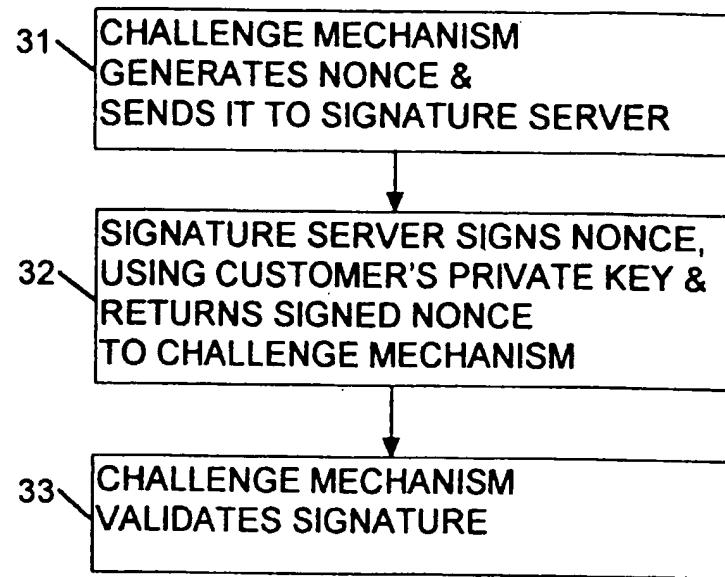


FIG. 3

- -----WARNING: INVALID SIGNATURE-----
42c02fd3ca73f38e1f1e--- WARNING: INVALID SIGNATURE -
- eea57d9d5001
- -----WARNING: INVALID SIGNATURE-----

The text area contains three instances of the message 'WARNING: INVALID SIGNATURE'. The first instance is preceded by a dash and followed by a horizontal line. The second instance is preceded by a dash and followed by a dash. The third instance is preceded by a dash and followed by a horizontal line.

FIG. 4

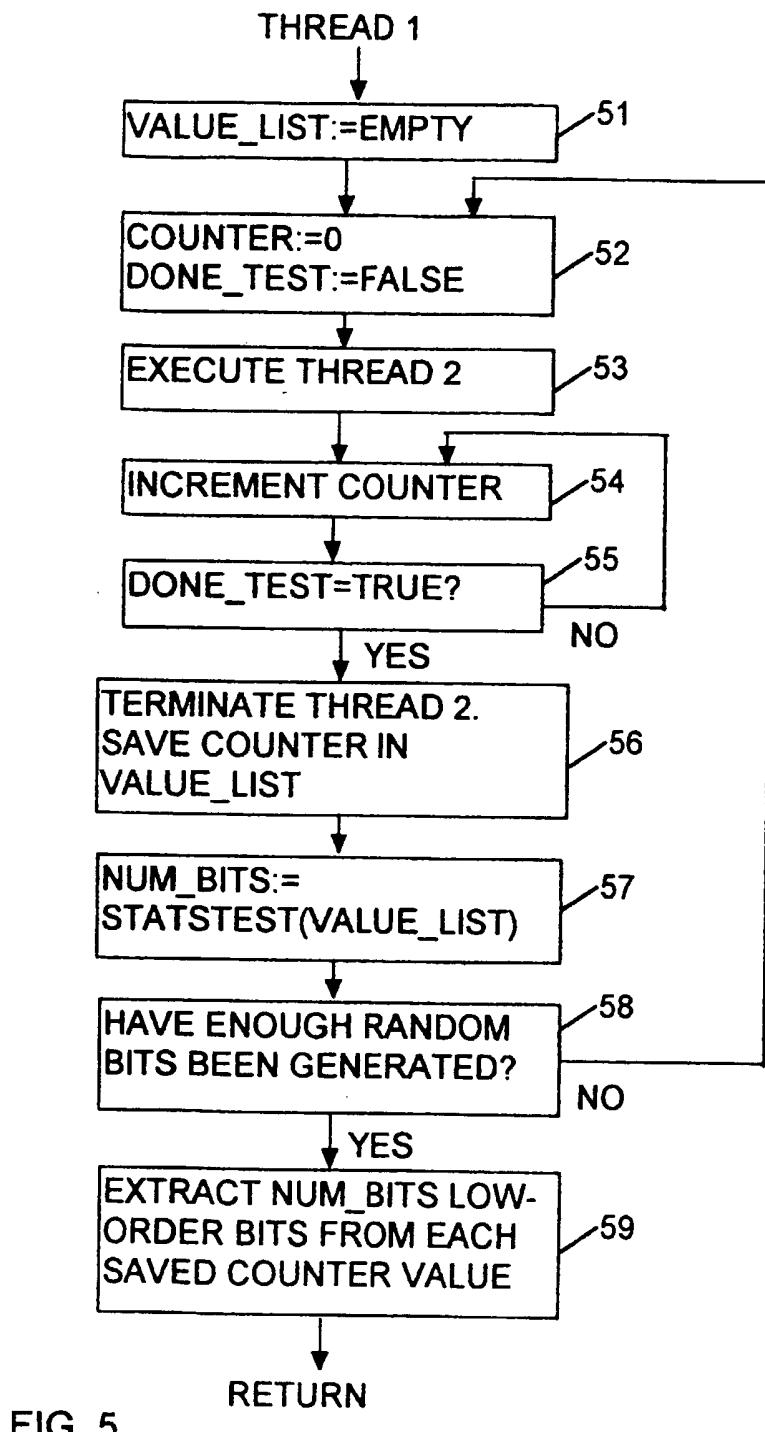


FIG. 5